Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2014) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

23rd International Meshing Roundtable (IMR23)

Degenerate Hex Elements

Clifton R. Dudley[a], Steven J. Owen[b,*]

[a]*Stanford University, Department of Mechanical Engineering, Stanford, California, 94305, U.S.A.*
[b]*Sandia National Laboratories*[1]*, Simulation Modeling Sciences, Albuquerque, New Mexico, 87185 U.S.A.*

Abstract

Automatic, all-hex meshes are required in many environments. However, current methods can produce unacceptable results where geometric features or topologic connectivity impose limiting constraints. Collapsing a small number of edges or faces in an all-hex mesh to produce degenerate hex elements may be sufficient to turn an otherwise unusable mesh into an adequate mesh for computational simulation. We propose a post-processing procedure that will operate on an existing all-hex mesh by identifying and collapsing edges and faces to improve element quality followed by local optimization-based smoothing. We also propose a new metric based upon the scaled Jacobian that can be used to determine element quality of a degenerate hex element. In addition we illustrate the effectiveness of degenerate elements in analysis and provide numerous meshing examples using the sculpt meshing procedure modified to incorporate degeneracies.

Peer-review under responsibility of organizing committee of the 23rd International Meshing Roundtable (IMR23).

Keywords: degenerate hexahedral elements, grid-based, overlay grid, smoothing, sculpt

1. Introduction

For computational simulation in many environments, analysts prefer the tri-linear 8-node hex element over its tetrahedron counterparts. In spite of significant overhead required to generate an all-hex mesh versus generation of a tet mesh of similar geometry, hex meshing remains an important requirement for many analysts. The ability to automatically generate a quality all-hex mesh for an arbitrary solid model has long been a major research challenge.

Many methods for fully automatic all-hex mesh generation have been proposed. We classify these as either *geometry-first* or *mesh-first* approaches. The geometry-first approaches, which may include algorithms such as mapping[1], sweeping[2], plastering[3,4], whisker weaving[5], and medial axis[6], involve developing a mesh using the CAD boundary representation as a framework from which to build the nodes and elements of a mesh to fill the geometric domain. Mesh-first approaches, such as overlay and octree grid-based methods[7–10] first construct a space-filling grid or mesh of nodes and elements. They then employ methods to locally modify the mesh to capture features of the geometry and topology of the CAD model.

[1]Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

*Corresponding author
E-mail address: sjowen@sandia.gov

Peer-review under responsibility of organizing committee of the 23rd International Meshing Roundtable (IMR23).

Where geometry-first methods such as sweeping can be successfully employed, they can result in high quality meshes. However, in practice, they usually necessitate user interaction, sometimes requiring heroic efforts to decompose the model into topologically consistent pieces before meshing. For automatic geometry-first methods such as plastering or whisker weaving, interior elements may have poor quality or closure may be impossible because of local topological constraints when attempting to close a void region with all hexahedral elements.

On the other hand, mesh-first methods, although having the potential of being completely automatic, can also sometimes suffer from poor element quality at the boundaries where geometric features and topologic constraints can be limiting.

The goal of generating quality all-hex meshes using geometry-first and mesh-first methods is laudable and deserves continued research. However in practice, current automatic methods can often produce results where the vast majority of elements are acceptable, with a small percentage that may fall below an acceptable threshold for analysis. In these cases, additional tedious, often manual work must be performed to clean up the mesh to improve quality, or they may be discarded all together. We propose a solution where we improve upon such meshes by incorporating a limited number of degenerate hex elements.

The concept of a degenerate element is not new. Indeed, some finite element implementations formulate triangles as degenerated quadrilateral elements. For example Graham et. al. [11] creates degeneracies through anisotropic refinement methods and their results show that degeneracies in the meshes yields no degradation in the approximation properties.

For 3D hex elements, Shelton et. al. [12] validate the use of degenerate hex elements in analysis. They provide an exhaustive numerical exploration of degenerated hexes and present results of patch tests that demonstrate linear completeness of degenerate elements. They also show optimal convergence rates for meshes containing degenerate elements and show applicability of degenerates for solving complex problems.

It is also notable that in Lipton et.al. [13], degenerate control meshes were explored for iso-geometric analysis using b-spline basis functions. They cite several cases of degeneracies where patch tests were exactly satisfied for all three polynomial orders.

Other meshing researchers have also encountered degenerate elements. However in most cases, strategies are devised to eliminate the occurrence of such elements. For example Schneiders [14] encounters degenerate elements in the course of his overlay grid algorithm, though attempts are made to get rid of them by splitting degenerate elements into a set of valid hexes. Taghavi [15] also attempts to eliminate degeneracies in his overlay grid procedure by introducing a limited number of pyramid, tet or wedge shaped elements at the mesh boundaries.

In this work, rather than eliminating degeneracies, we propose strategically introducing a limited number of degenerate hexes to improve an otherwise unusable or poor quality mesh. This can be accomplished by identifying poor quality elements and performing edge and face collapse operations to define a few degenerate hex elements within the mesh. At first glance, it may seem that collapsing edges on a hex element would be detrimental to the mesh quality and subsequent solution accuracy. It has been demonstrated [12], however that incorporating a limited number of degenerate hex elements in a mesh causes little effect on the solution accuracy and in some instances can even improve results.

To demonstrate the effectiveness of degenerate elements, we expand upon the author's work, *sculpt*[10,16,17]. Application of degenerate elements to sculpt meshes proves especially useful since it can be applied as a post process procedure to an existing mesh where a small percentage of elements near the boundaries can be improved through degeneration.

Formation of degenerate hex elements could be equally applied to advancing front hex procedures such as plastering and whisker weaving where fronts collide and topological constraints prohibit formation of all-hex elements. These methods would, however, require customized rules for element formation based on local topological constraints in order to form usable degenerate hex elements. Indeed, initial implementations of Plastering utilized wedges [3] and knife [18] shapes, forms of degenerate hexes, to resolve interior voids. Recent experiments to extend this effort to incorporate more general forms of degeneracies proved problematic. For this reason we utilize sculpt as the target application, using edge and face collapses to define degeneracies on an existing mesh. The proposed work could be equally applied to any existing mesh that would benefit from local mesh quality improvement through this same technique.

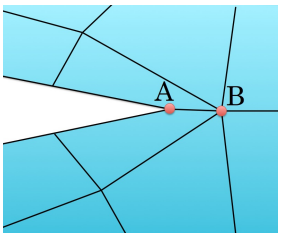


Fig. 1. Sharp geometric features can result in poor mesh quality near boundaries as evidenced by the elements adjacent edge AB.

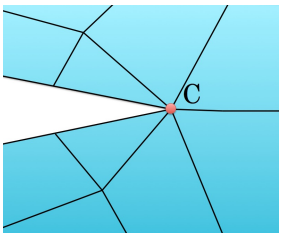


Fig. 2. Edge AB has been collapsed into C and results in degenerate (triangle) elements but improved quality.

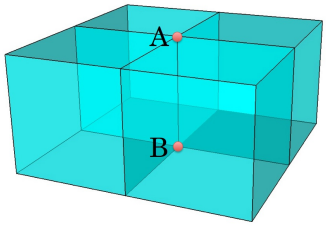


Fig. 3. Initial set of hexes sharing edge AB

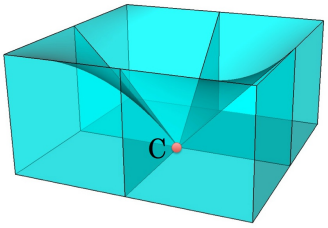


Fig. 4. Edge AB has been collapsed into C and results in degenerate (rock) elements.

We cite one example where edge collapsing may be useful in the context of overlay grid meshing procedures. Geometric sharp features and high curvature can often be a challenge for sculpt, and overlay grid methods in general. Figure 1 shows a 2-dimensional example where the boundary layer must navigate a sharp feature resulting in poor mesh quality. Figure 2 however shows the same local mesh where edge AB has been collapsed into a single node C. Although some of the resulting elements are now triangles, the local mesh quality at the feature is improved. This same concept can be extended to 3-dimensions where the quality of the hex elements surrounding a feature may be poor. While in 2D, the resulting degenerate configuration can only be a triangle, an edge collapse operation, such as that shown in Figures 3 and 4 will generate a set of *rock*-shaped elements. Further edge or face collapses can result in a wide variety of possible shapes.

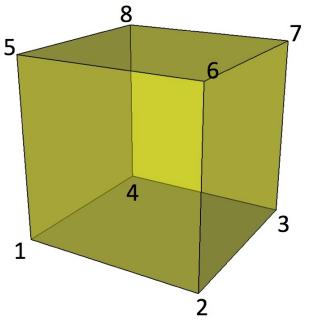
In this work, we first outline the definition of degenerate hex elements and their acceptable forms. We also propose a method for computing a modified scaled Jacobian metric for these new element shapes. We later propose an automatic method for identifying poor quality elements and improving them through targeted edge and face collapse operations followed by local mesh optimization. Examples of meshes generated using sculpt where this algorithm has been employed are presented and their mesh quality examined. Finally we use meshes that incorporate degeneracies in a computational simulation and compare their accuracy to meshes that do not include degeneracies.

2. Degenerate Hexes

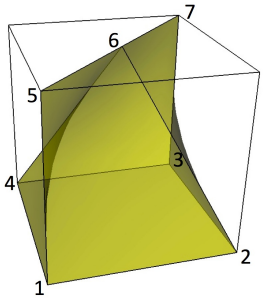
2.1. Degenerate Hex Classification

A standard eight node hexahedron element can be defined in terms of the connectivity of its nodes. Indeed, many FEA file formats [19] first prescribe a list of ordered nodes in the model, followed by the connectivity of each individual hex element defined by the 8 ordered numerical IDs of its nodes. The order and orientation of the nodes in a standard hex is shown in table 1 (1). To avoid defining a unique element type for each permutation of degeneracy, we can describe each by simply repeating nodes of the standard hex. Table 1 shows 13 different permutations of degenerate hex shapes constructed by repeating nodes in the standard hex. This has the advantage of utilizing existing FEA formats and can be readily incorporated into analysis codes without extensive restructuring for new element types.

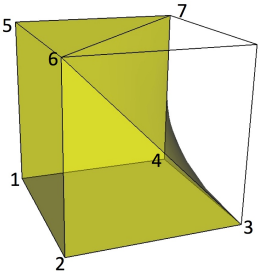
The element shapes shown in Table 1 are constructed by progressively collapsing edges or faces in the standard hex. Also shown for each degenerate shape is an example connectivity, indicating where node IDs would be repeated.



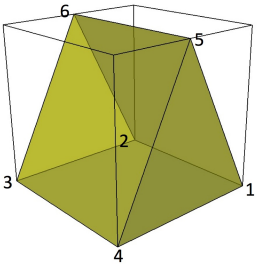
(1) Standard Hex
8 nodes, 6 quads
1-2-3-4-5-6-7-8



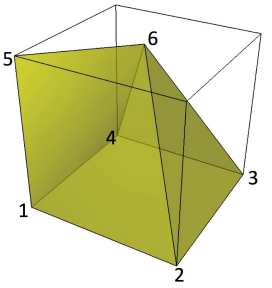
(2) Knife
7 nodes
5 quads
1-2-3-4-5-6-7-6



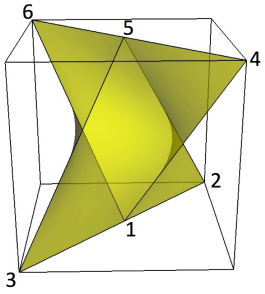
(3) Rock
7 nodes
4 quads, 2 tris
1-2-3-4-5-6-6-7



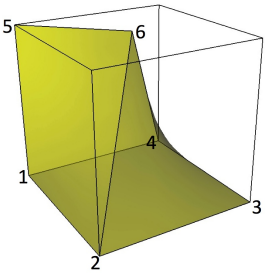
(4) Wedge
6 nodes
3 quads, 2 tris
1-2-3-4-5-6-6-5



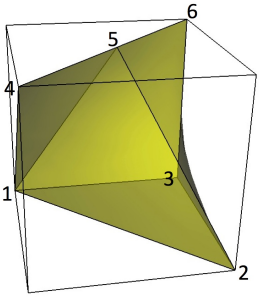
(5) Axe
6 nodes
3 quads, 2 tris
1-2-3-4-5-6-3-6



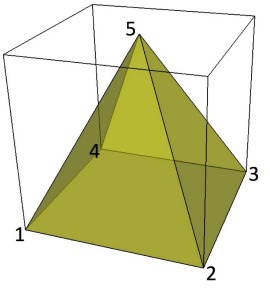
(6) Double Knife
6 nodes
4 quads
1-2-1-3-4-5-6-5



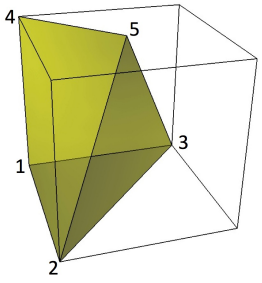
(7) Hyper Axe
6 nodes
4 quads
1-2-3-4-5-6-2-6



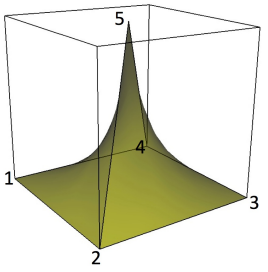
(8) Hyper Knife
6 nodes
3 quads, 2 tris
1-2-2-3-4-5-6-5



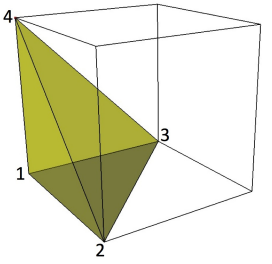
(9) Pyramid
5 nodes
1 quad, 4 tris
1-2-3-4-5-5-5-5



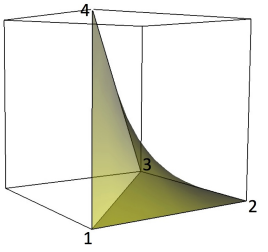
(10) Half Knife
5 nodes
2 quads, 2 tris
1-2-2-3-4-5-2-5



(11) Needle
5 nodes
3 quads
1-2-3-4-2-5-2-5



(12) Standard Tet
4 nodes
4 tris
1-2-2-3-4-4-2-4



(13) Hyper Tet
4 nodes
1 quad, 2 tris
1-1-2-3-1-4-1-4

Table 1. Degenerate hex element configurations. Shows 13 acceptable element shapes that can be used as finite elements represented as degenerate hexes. Example connectivity for each element shape is shown indicating repeated nodes in the element connectivity to form the degeneracies.

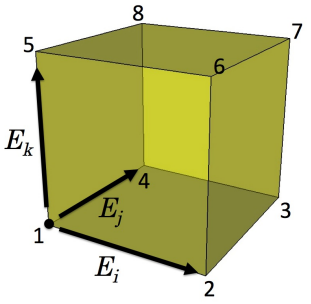


Fig. 5. Ordered edges E_i, E_j, and E_k are used to compute the scaled Jacobian at node 1

We note that these node numberings are however not unique. The rock element shape, for example, has 12 different potential node numberings, one for each possible edge collapse of the standard hex. We also note that Table 1 is not exhaustive of all shapes that can be formed by edge and face collapses from a hex element. However the shapes illustrated represent those shapes that are most likely to be useful alternatives to an otherwise poor quality hex element. Table 1 can serve as a pool of elements from which a meshing procedure may identify and select shapes to assist in improving the quality of an existing mesh.

2.2. Mesh Quality of Degenerate Hexes

For our purposes, we define acceptable quality in terms of the minimum scaled Jacobian, J_s of the element. The eight scaled Jacobian values, $(J_s)_I$ at the nodes of a standard hex can be computed by taking the determinant of its three ordered normalized edge vectors $E_{i,j,k}$ as illustrated in Figure 5 and Equation (1). The scaled Jacobian metric for a hex is then taken as the minimum of the eight determinant calculations as in Equation (2).

$$(J_s)_I = det\left\{\hat{E}_i\hat{E}_j\hat{E}_k\right\}^{\top} \tag{1}$$

$$J_s = min\left((J_s)_I, I = 1, 2, ...8\right) \tag{2}$$

A value of $J_s = 1.0$, indicates an ideal element where all angles are precisely 90 degrees, however a value of $J_s \leq 0.0$ normally indicates an unacceptable element for computational purposes. Depending on the requirements of the analysis, an acceptable value for scaled Jacobian can vary, but normally a value of $J_s \geq 0.2$ is permissible.

Equation (1) requires three edges at a node, yet as can be seen in Table 1, valid degenerate hex cases can have 2, 3 or 4 edges connected to each node. In developing our metric for these degenerate nodes, we expand on the quality calculation techniques proposed by Knupp [20]. Just as non-simplicial elements (hexes, quads) require multiple evaluations of nodal Jacobians to determine the hex Jacobian, our method considers multiple combinations of Equation (1) at non 3-valent nodes to determine the minimum equivalent value for scaled Jacobian. For example, the 7-node knife element, shown in Figure 6, node 6 has a valence of four and nodes 5 and 7 have valences of two. To accommodate a 4-valent node we propose computing the scaled Jacobian metric for the four permutations of three edges at the node and using the minimum of the four as the node's contribution to Equation (2). This is illustrated in Figure 6 where the four different permutations of E_i, E_j, E_k at node 6 and where right hand rule is maintained, are shown. To accommodate a 2-valent node such as node 5 in Figure 6, we treat the 2-valent node as a 4-valent using the quad diagonals as the two additional vectors. Figure 6 also illustrates the 4 permutations of E_i, E_j, E_k that would be used at node 5 to compute $(J_s)_I$. For other 3-valent nodes in the element, the standard contribution to the element J_s can be computed using Equation 1.

In designing the scaled Jacobian metric for degenerate hex elements, we extend the standard practice of taking the minimum scaled Jacobian at a single node by incorporating additional measures based on local nodal valence. While we concede that this solution is somewhat heuristic, our results indicate that these measures provide resonable consistency with standard hex elements in practice. For example, when used within optimization-based smoothing, they tend to favor dihedral angles closer to ninety degrees regardless of local node valence as well as increasing volume. Invalid cases, where scaled Jacobian drops below zero, can also be quickly identified.

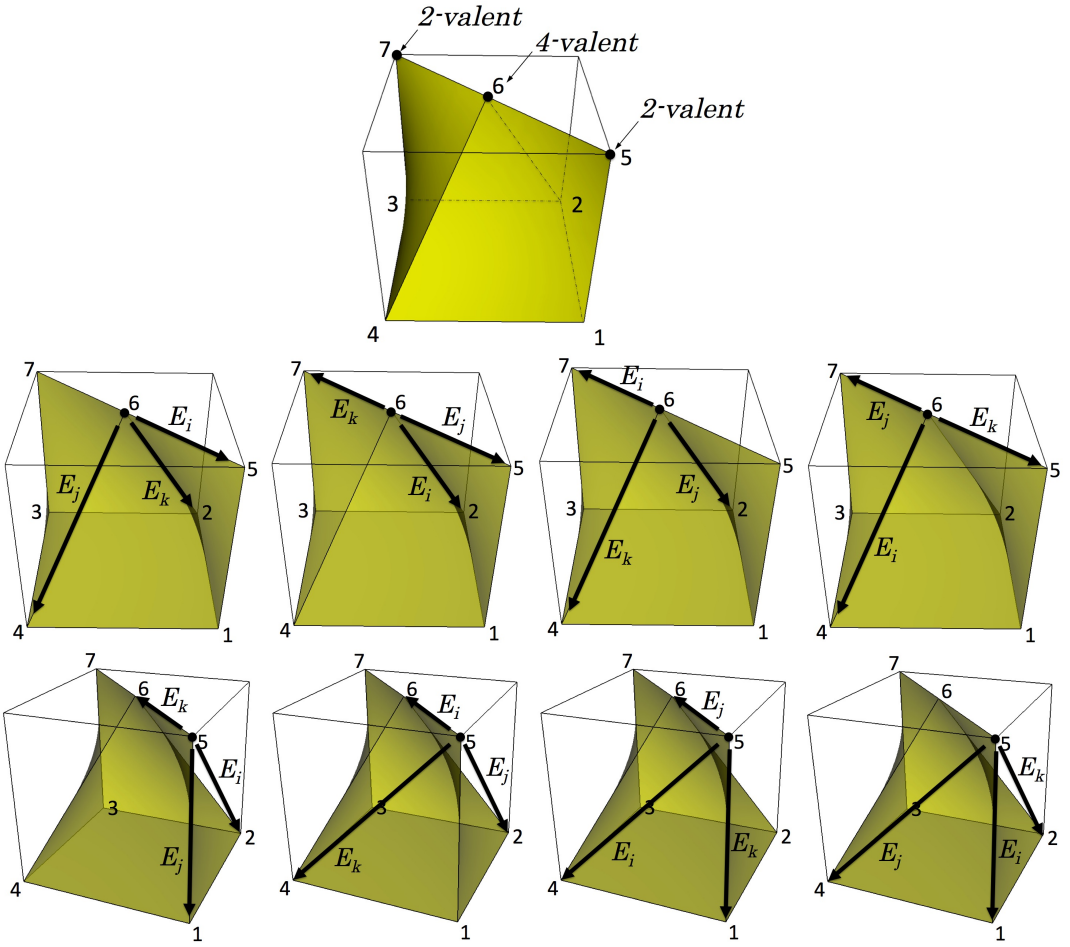


Fig. 6. Calculation of the scaled Jacobian of a knife element. $(J_s)_I$ for node 6, a 4-valent node, is computed as the minimum determinant of the 4 permutations of E_i, E_j, E_k shown here. The 2-valent nodes (5 and 7) utilize the quad diagonals to define 4 permutations of E_i, E_j, E_k.

We note that Equation (2) is effective in measuring and controlling dihedral angles in the element, however, aspect ratio and edge lengths are not satisfactorily managed. To control for these factors we include a size scale factor S_f on the scaled Jacobian as shown in Equation 3.

$$(J_s)_I = S_f det\left\{\hat{E}_i \hat{E}_j \hat{E}_k\right\}^{\top} \tag{3}$$

$$S_f = \begin{Bmatrix} e_s \leq S_t, & \frac{e_s}{S_t} \\ e_s > S_t, & \frac{S_t}{e_s} \end{Bmatrix} \tag{4}$$

$$e_s = min(\|E_i\|, \left\|E_j\right\|, \|E_k\|) \tag{5}$$

where S_t is a target edge size. In this case, we define S_f as the constant size of one cell of the Cartesian grid used for the sculpt overlay grid. The modified scaled Jacobian that uses Equation 3 to 5 has the effect of favoring elements that meet the size criteria as well as the angle criteria.

2.3. Degenerate Hex Construction

We propose an algorithm for automatically improving a hex mesh by identifying poor quality elements and collapsing edges or faces to create degenerate hexes. To begin, we first select only those elements that fall below a

	Model	Num. Elems	Num. Degen.	Initial J_s	J_s w/Degenerates	J_s Improvement
A	a_trol02-6	374,454	262	-0.545849	-0.140418	0.405431
B	spindle-8	43,723	18	-0.228354	0.0795396	0.3078936
C	rib2a-10	11,749	66	-0.145585	0.12349	0.269075
D	pl_mold2-10	10,035	16	-0.154436	0.0773877	0.2318237
E	thinwedge-6	358,036	3	0.187658	0.378927	0.191269
F	keg_ds-6	229,343	75	0.0285279	0.198239	0.1697111
G	axel-8	61,826	3	0.360202	0.438501	0.078299
H	valvola-8	42,210	6	0.405991	0.450391	0.0444
I	railsupport-10	8,644	0	0.200045	0.200045	0.0
J	pipe-6	405,596	126	0.191156	0.128739	-0.062417

Table 2. A sampling of results from Sculpt meshes. Shows scaled Jacobian J_s, before and after insertion of degenerate hexes and the associated quality improvement. Note that J_s represents the lowest quality element in the mesh

user-defined J_s threshold. In most cases a value $Js < 0.2$ is considered poor quality and is a candidate for degeneration. From those elements we identify candidate edge or face collapses and perform only those collapses that would result in an improved element quality of the minimum J_s for all attached hexes or degenerate hexes.

To ensure consistency, the poorest quality elements are first sorted according to J_s, and the worst quality element is processed first. Element quality of the hex and its immediate neighbors are evaluated for each candidate edge or face collapse. Of the candidate edge or face collapse operations at the hex, we identify the edge or face associated with the best improvement to the minimum mesh quality and perform that collapse. If all candidate edge or face collapses for a given hex will not result in improved minimum element quality for the hex and its neighbors, then the element is left as-is. After each collapse, the element and its neighbors are reevaluated and resorted and we continue to process the next-to-worst element. This procedure continues until candidate collapse operations for all hexes in the queue will no longer improve minimum mesh quality.

Following each collapse, a local optimization based smoothing procedure [17] is used to position the new consolidated node resulting from the collapse. This procedure utilizes Equation 3 as the objective function. It operates to improve the worst quality element at the node based upon the new element connectivity resulting from the edge collapse. Further smoothing of adjacent nodes may also be advantageous to further improve element quality, however our initial implementation and results were limited to smoothing only the collapse node. Note that attached elements may be either standard hexes or a combination of any of the degenerate hex shapes illustrated in Table 1. In practice, however, we see most commonly the 7-node rock and knife shapes formed as a result of this procedure.

3. Results

3.1. Meshing with Degenerate Hexes

To test the effectiveness of our algorithm, we tested over 100 single-part CAD models. We used sculpt to mesh each of the CAD models at three different mesh resolutions. To get a base-line, we first meshed all of the models without use of automatic edge and face collapses. We then meshed all of the same models again using automatic edge and face collapses and a threshold of $J_s < 0.2$ scaled Jacobian. We illustrate a representative sample in Figure 7 and their results in Table 2.

From these results we see that in most cases a very small percentage of elements, typically well below 0.1% are converted to degenerates, when compared to the total number of elements in the model. In many cases, the small change is sufficient to increase the minimum element quality from where the mesh would be unusable to being useful for computation. Figure 8 shows the distribution of degenerate elements in the (C) rib2a-10 model. We observe that clusters of degenerate hexes are typically distributed throughout the model, but most frequently at boundaries where high curvature or sharp features are present. We note that in most cases an increase in mesh quality is achieved,

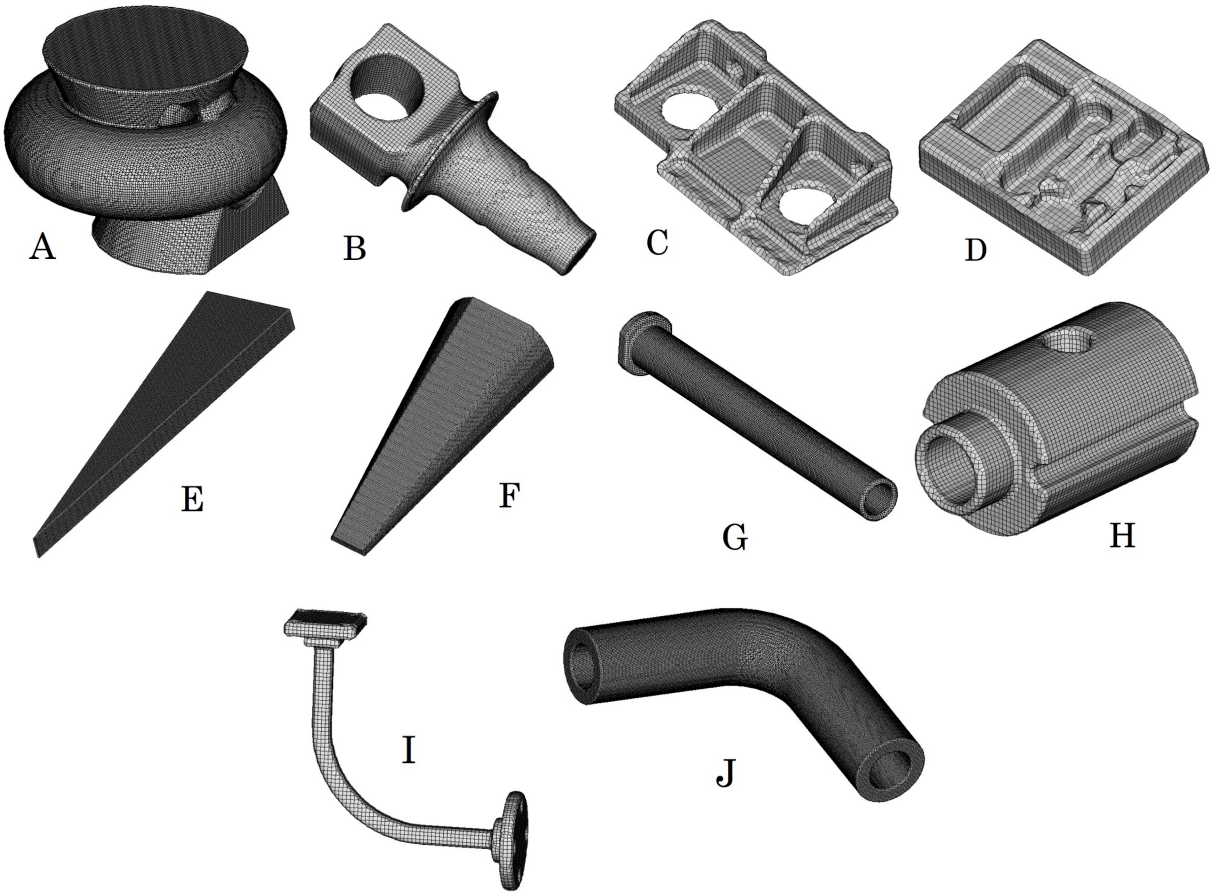


Fig. 7. A sampling of the meshes generated using Sculpt and automatic formation of degenerate hexes

however the current state of the algorithm does not guarantee a computable mesh in all cases. Indeed, a few rare cases showed a decrease in element quality.

A summary of the results from a total of 447 separate test cases is shown in Tables 3 and 4. Table 3 indicates a statistically modest increase in the minimum mesh quality of the 447 test cases when using degenerate elements. For example, without degenerates, approximately 87.9% of the models would mesh with $J_s > 0.0$. This increases to 90.1% when incorporating degenerates. Similarly, the percentage of models with $J_s > 0.2$ increases from 61.1% to 72.0%.

Table 4 indicates that of all cases we tested, 53.2% actually required insertion of degenerate elements to improve element quality. The remaining meshes either already had adequate element quality, or insertion of degenerates would not improve results. Of those 53.2%, 79.0% achieved an increase in element quality. We also noted that approximately 7.1% of the cases suffered a modest decrease in element quality. Some initial investigation into the cause of reduced element quality points to smoothing procedures that will need to be improved. The ability to locally untangle a mesh that incorporates degenerate hexes is still a work in progress.

	Without Degenerates		With Degenerates	
	Num. Models	% Models	Num Models	% Models
Min $J_s > 0.0$	393	87.9	403	90.1
Min $J_s > 0.2$	273	61.1	322	72.0

Table 3. Shows the minimum mesh quality with and without the creation of degenerate hexes. Based on 477 total meshes generated with sculpt [17]

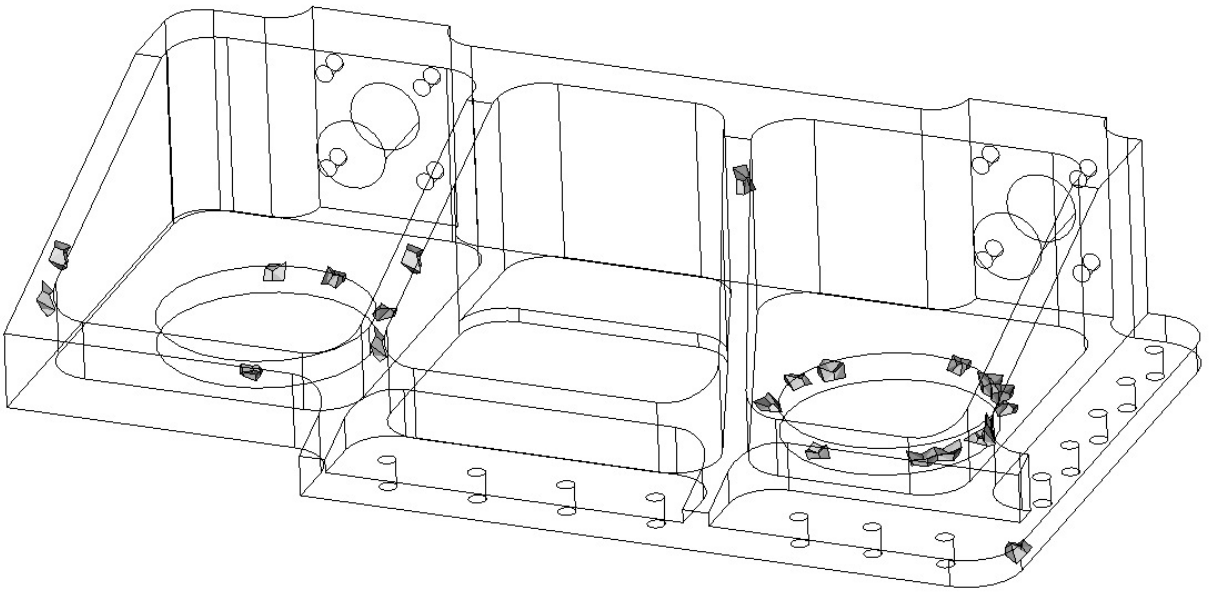


Fig. 8. rib2a-10 model showing the distribution of its 66 degenerate hex elements throughout the mesh.

	No. Models	% Models
Total Sculpted	447	100
Total w/ Degenerates	238	53.2
Number improved	188	79.0
Number worsened	17	7.1

Table 4. Summary of mesh quality results from 477 sculpt meshes.

3.2. Performance of Degenerate Hexes in Analysis

To validate the performance of degenerate elements in sculpt meshes we extend the study previously published by the author in [16]. For this study we chose a simple torsion pin shown in Figure 9. In this case we model a pin fixed to a rigid body. We apply a rotational displacement to the end of the pin and measure the integrated torque reaction at the rigid body. Analysis is performed using the explicit quasi-static code, Sierra Solid Mechanics [21] using a linear elastic material model. Rotational displacement is applied over a 1 second time period up to a 15 degree rotation. The torsion pin itself has a step down in radius at its center recognizing that stress concentrations will develop at the re-entrant corner that must be handled by the sculpt mesh and any degenerate hexes formed.

The sculpt meshes displayed in Figure 10, show three different orientations of the base Cartesian grid. Although presumably ideal to align the base Cartesian grid with the main orientation of the geometric model, there is no guarantee that this can be accomplished in practice. As a result, we will look at the sensitivity of the final solution to the orientation of the base Cartesian grid. In this case we choose 10 degree increments of the Cartesian grid up to 90 degrees.

For the initial study, described in [16], the reference solution was computed by performing a convergence study on a progressively refined mesh. We compared the mesh generated with sculpt to those generated with traditional pave-and-sweep methods. We extend that study by utilizing sculpt and incorporating degenerate elements as defined by the algorithm in this work.

Table 5 and Figure 11 illustrate the results of the study. Of particular note in this study is the observation that including degenerate elements in the mesh resulted in almost no change to the overall solution when compared to not using them. This indicates that use of degenerate elements in solution is not detrimental to the accuracy of the

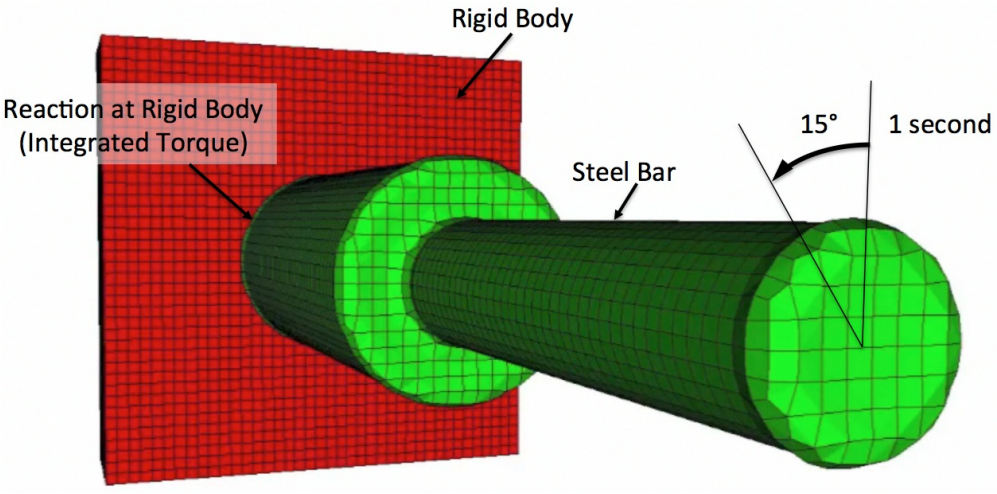


Fig. 9. Sculpt mesh of torsion pin used in linear elastic simulation

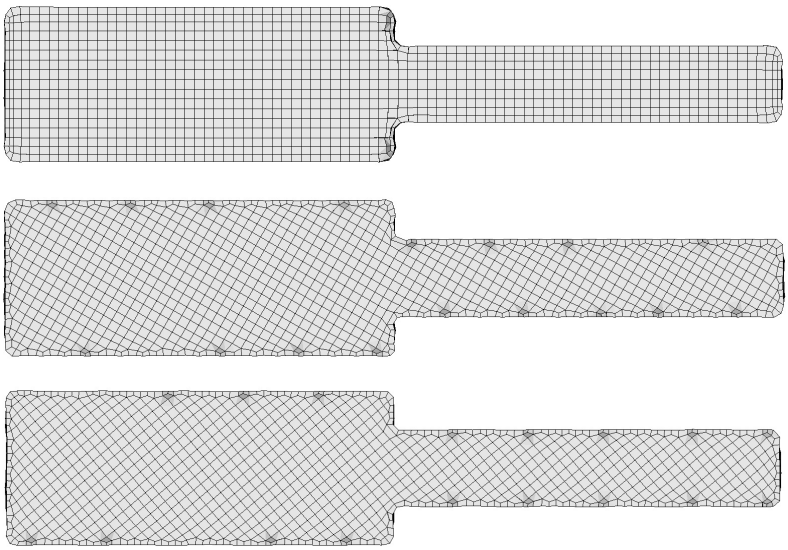


Fig. 10. Cutaway view of Sculpt mesh of torsion pin showing grid oriented at 0, 30 and 50 degrees respectively.

solution. Indeed we observe that the effects of changes in the orientation of the overlay grid are much more significant than the effects of degeneracies in the mesh, although rotational effects are still within a range of 1%.

4. Conclusions

In this work we have proposed using a limited number of degenerate hex elements to transform an otherwise unusable mesh into a satisfactory mesh for computational simulation. We have introduced a classification of valid degenerate shapes and proposed a metric based on the scaled Jacobian to evaluate their quality. A method for automatically creating degenerate elements from an existing mesh by the use of edge and face collapses was introduced and the method was evaluated using a wide range of CAD models. Finally the performance of degenerates in a quasi-static analysis was evaluated and compared to the same model without degenerates.

Test Case	Num. Elems	Num. Degen.	Min J_s All-Hex	Min J_s w/Degen.	Percent Error All-Hex	Percent Error w/Degen.
Sculpt-00	13578	6	0.211406	0.324209	2.23229%	2.22490%
Sculpt-10	14083	24	0.355677	0.369437	1.68043%	1.67992%
Sculpt-20	14313	33	0.290767	0.315983	1.55861%	1.56012%
Sculpt-30	14501	81	0.300514	0.34585	1.33490%	1.33445%
Sculpt-40	14607	84	0.280919	0.280919	1.07916%	1.07673%
Sculpt-50	14607	87	0.298308	0.330422	1.08264%	1.08068%
Sculpt-60	14501	96	0.103696	0.338327	1.33114%	1.33071%
Sculpt-70	14313	30	0.333133	0.333133	1.57603%	1.57484%
Sculpt-80	14083	12	0.354224	0.369439	1.67958%	1.67915%
Sculpt-90	13575	9	0.203566	0.342928	2.30085%	2.30525%

Table 5. Results from torsion analysis study from 10 different sculpt meshes oriented at 10 degree intervals. Results are shown for meshes with and without degenerate elements

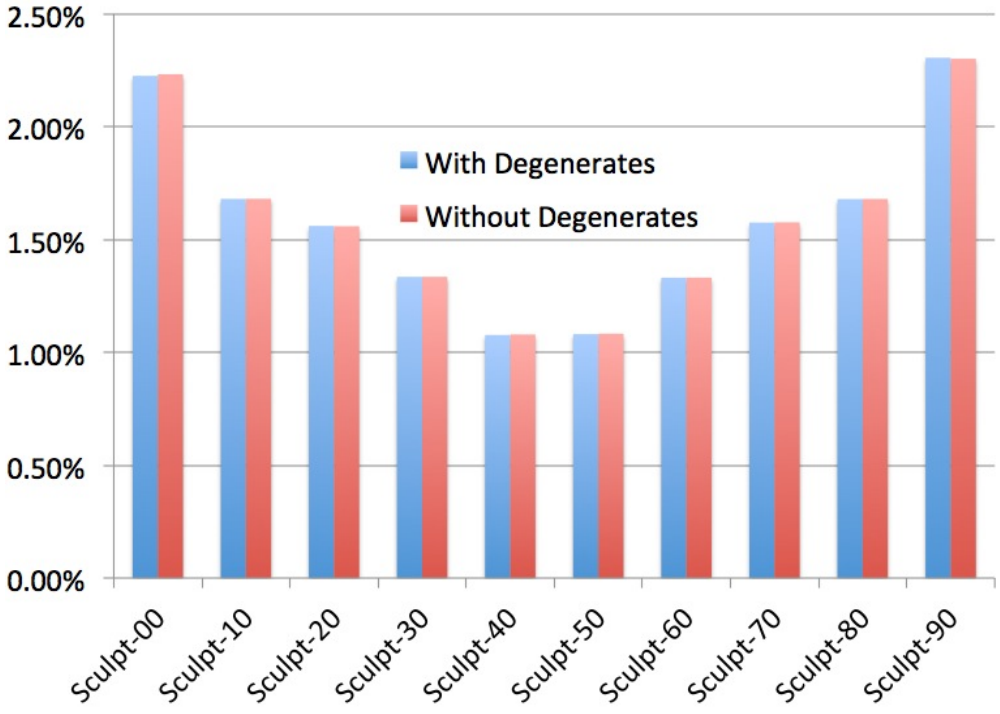


Fig. 11. Comparison of percent error between meshes with and without degenerates. Note that observed results between two cases are almost identical.

We noted that differences between meshes that incorporated a limited number of degenerate hex elements performed almost identically to those without. This observation alone is significant, as it indicates that an all-hex mesh is not necessary to achieve satisfactory results. These results have major implications on mesh generation algorithm development that has for many years focussed on creating a quality, all-hex mesh.

The method we have proposed for creation of degenerate elements is still a work in progress. The current results show a statistically significant increase in element quality over the range of CAD models tested. We have however, identified specific issues that can be improved in the course of this study and will continue to make adjustments. We recognize the need to better incorporate degenerate elements in an optimization-based smoothing scheme that includes untangling of negative Jacobian elements, a factor that may be causing some observed deterioration in element quality in a few cases. Additionally the modification of the algorithms to be used in a parallel message passing environment

must be addressed. Higher degree forms of degenerate elements were also not studied and would be an interesting extension to this work.

We recognize that use of degenerate hex elements in analysis may not be supported in many analysis codes. The authors have the benefit of working directly with the developers of the Sierra Mechanics analysis tools [21] who have made appropriate modifications to support degenerate elements. We note however, that these modifications [12] were minimal as in most cases they can be treated as a standard linear hex with repeated nodes in their connectivity. In order for the use of degenerate elements to continue to gain acceptance, it will be necessary for major vendors and developers of analysis tools to support the use of degenerates. The benefits however, would be substantial, where users could take advantage of fast, automatic hex meshing technologies reducing the need for manual methods that currently require time-consuming decomposition and geometry clean-up.

4.1. Acknowledgments

Special thanks goes Tim Shelton who provided the computational tools and the theoretical motivation [12] for the use of degenerate hex elements in solid mechanics. Our thanks also go to Brett Clark and Christopher Kozuch for their reviews and input on the paper. We acknowledge funding for this work through the U.S. Department of Energy's Advanced Scientific Computing Program.

References

[1] W.A. Cook, W.R. Oakes, Mapping methods for generating three-dimensional meshes, Computers in Mechanical Eng., Aug (1982) 67–72
[2] M.L. Staten, S.A. Canann, S.J. Owen, BMSWEEP: Locating interior nodes during sweeping, Proc. 7th Int. Meshing Roundtable (1998) 7–18
[3] T.D. Blacker, R.J. Meyers, Seams and wedges in plastering: A 3D hexahedral mesh generation algorithm, Eng. with Computers, 2(9) (1993) 83–93.
[4] M.L. Staten, R.A. Kerr, S.J. Owen, T.D. Blacker, Unconstrained paving and plastering: progress update, Proc., 15th Int. Meshing Roundtable (2006) 469–486
[5] T.J. Tautges, T.D. Blacker, S.A. Mitchell, The whisker weaving algorithm: A connectivity-based method for constructing all-hex meshes, Int. J. Numer. Meth. Engrg., 39 (1996) 3327–3349
[6] T.K.H. Tam, C.G. Armstrong, 2D finite element mesh generation by medial axis subdivision, Advances in Eng. Software, 13 (1991) 313–324
[7] R. Schneiders, F. Schindler, F. Weiler, Octree-based generation of hexahedral element meshes, Proc. 5th Int. Meshing Roundtable, (1996) 205–216
[8] Y. Zhang, C.L. Bajaj Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. Comp. Methods in Applied Mechanics and Eng. 195 (2006) 942–960
[9] Y. Ito Y, A.M. Shih, B.K. Soni, Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates, Int. J. Num. Meth. Engrg, 77(13) (2009) 1809–1833
[10] S.J. Owen, M.L. Staten, M.C. Sorensen, Parallel hex meshing from volume fractions, Proc. 20th Int. Meshing Roundtable, (2011) 161–178
[11] I.G.Graham, W. Hackbusch, S.A. Sauter, Finite elements on degenerate meshes : inverse-type inequalities and applications, IMA Journal of Numerical Analysis, 25 (2005) 379–407
[12] T.R. Shelton, N.K. Crane, J.V. Cox, An exploration of accuracy and convergence of the degenerate uniform strain hexahedral element: A solution to the unmeshed void in an all-hexahedral mesh, Proc. ASME 2013 Int. Mechanical Eng. Congress and Exposition, (2013) Paper No. IMECE2013-63270
[13] S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj, T.J.R. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Comput. Methods Appl. Mech. Engrg. 199 (2010) 357–373
[14] R. Schneiders, A grid-based algorithm for the generation of hexahedral element meshes, Engrg. with Computers, 12(3-4) (1996) 168–177
[15] R. Taghavi, Automatic, parallel and fault tolerant mesh generation from CAD, Engrg. with Computers, 12(3-4) (1996) 178-185
[16] S.J. Owen, T.R. Shelton, Validation of grid-based hex meshes with computational solid mechanics, 22nd Int. Meshing Roundtable, (2013) 39–56
[17] S.J. Owen, Parallel smoothing for grid-based methods, 21st Int. Meshing Roundtable, Research Note (2012)
[18] B.W. Clark, S.E. Benzley, Development and evaluation of a degenerate seven-node hexahedron finite element, 5th Int. Meshing Roundtable (1996) 321–332
[19] G. Sjardema, ExodusII, http://sourceforge.net/projects/exodusii/ (2014)
[20] P. Knupp, Algebraic mesh quality metrics, Siam J. Scientific Comput., 23(1) (2001) 193–218
[21] Sierra Solid Mechanics Team, Adagio 4.22 User's Guide, Sandia National Laboratories, (2011) Sandia Report SAND2011-7597